

# **EXHIBIT 1**

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

The following table sets forth each limitation of each asserted claim, together with the feature of the Accused Product that Defendant Google LLC ("Google") makes, uses, sells, sold, and/or offers for sale and meets the limitation. Plaintiff InfoGation Corporation ("InfoGation") reserves the right to supplement or otherwise modify this preliminary claim chart as additional information becomes available during the course of discovery and after an entry of the Court's Markman Order.

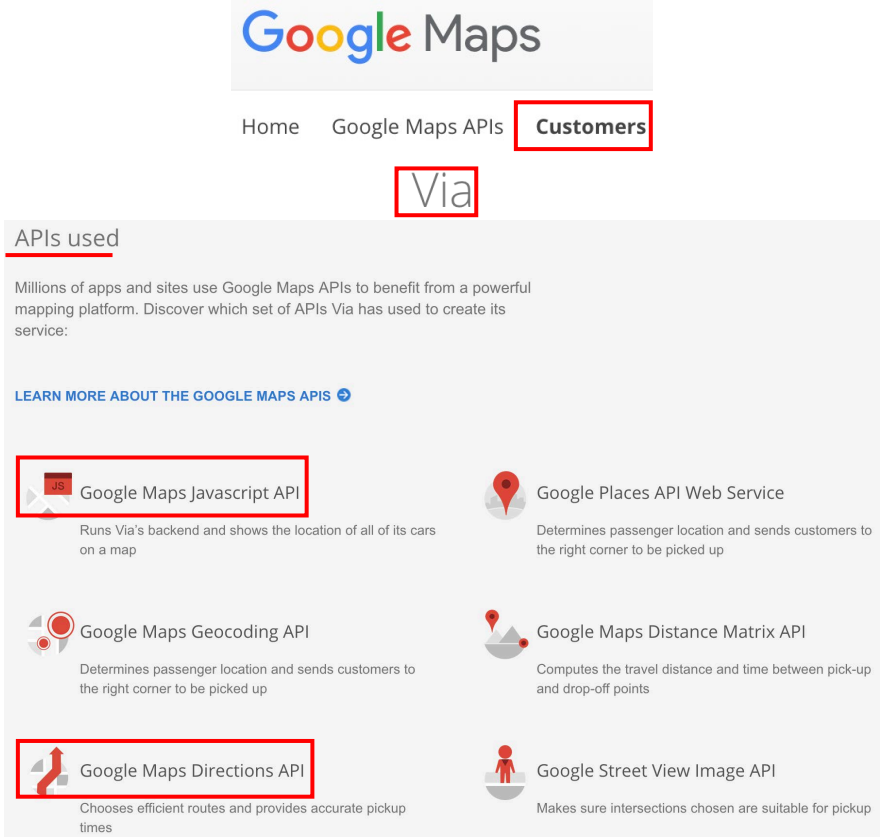
Claim Limitation	Accused Product
21. A method for providing an optimal route using real-time information from a server of a navigation system,	<div><p>The screenshot shows the Google Maps 'Customers' page. At the top, the Google Maps logo is visible, followed by navigation links: 'Home', 'Google Maps APIs', and 'Customers' (which is highlighted with a red box). Below the navigation bar, the 'Via' logo is displayed in a red box. The main content area is titled 'APIs used' and describes how millions of apps use Google Maps APIs. A link 'LEARN MORE ABOUT THE GOOGLE MAPS APIS' is provided. Below this, six API cards are shown, each with a red box around its title: 'Google Maps Javascript API' (described as running Via's backend), 'Google Places API Web Service' (determines passenger location), 'Google Maps Geocoding API' (determines passenger location), 'Google Maps Distance Matrix API' (computes travel distance), 'Google Maps Directions API' (chooses efficient routes), and 'Google Street View Image API' (makes sure intersections are suitable). A URL <a href="https://enterprise.google.com/maps/customers/via.html">https://enterprise.google.com/maps/customers/via.html</a> is provided. At the bottom, the 'Google Maps Platform' navigation bar is visible, with 'Web' selected and 'Maps JavaScript API' highlighted in a red box. Below this, the 'Directions Service' section is shown, including an 'Overview' subsection. The overview text states: 'You can calculate directions (using a variety of methods of transportation) by using the <code>DirectionsService</code> object. This object communicates with the Google Maps API Directions Service which receives direction requests and returns an efficient path. Travel time is the primary factor which is optimized, but other factors such as distance, number of turns and many more may be taken into account. You may either handle these directions results yourself or use the <code>DirectionsRenderer</code> object to render these results.' A URL <a href="https://developers.google.com/maps/documentation/javascript/directions">https://developers.google.com/maps/documentation/javascript/directions</a> is provided at the bottom.</p></div>

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM


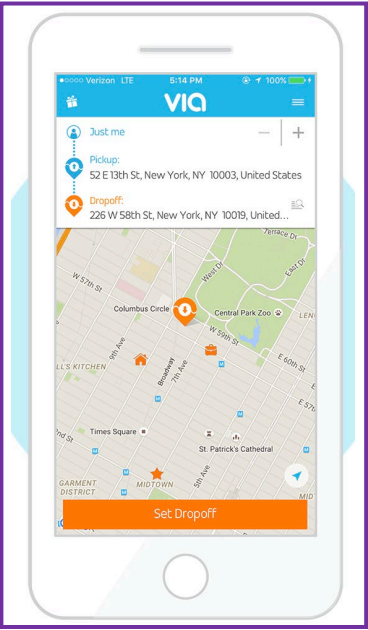
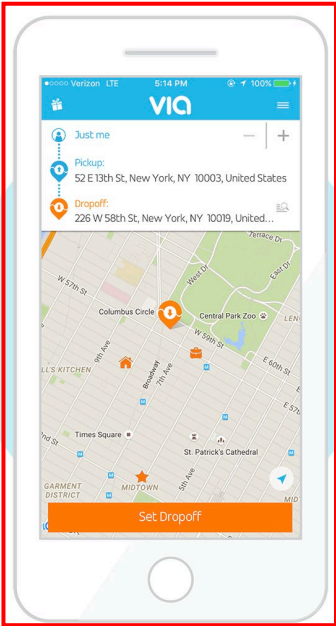

	<div></div> <div><h3>Routes</h3><p>Help your users find the <u>best way to get from A to Z</u> with <u>comprehensive data</u> and <u>real-time traffic</u>.</p><p><a href="https://cloud.google.com/maps-platform/">https://cloud.google.com/maps-platform/</a></p></div> <div></div> <div><p><a href="https://ridewithvia.com/">https://ridewithvia.com/</a></p></div> <div><h3>Directions Requests</h3><p>Accessing the Directions service is asynchronous, since <u>the Google Maps API needs to make a call to an external server</u>. For that reason, you need to pass a <i>callback</i> method to execute upon completion of the request. This callback method should process the result(s). Note that the Directions service may return more than one possible itinerary as an array of separate <code>routes[]</code>.</p><p><a href="https://developers.google.com/maps/documentation/javascript/directions">https://developers.google.com/maps/documentation/javascript/directions</a></p></div>
<p>the navigation system also comprising a client and said server coupled to a computer network, said method comprising the steps of:</p>	

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM



<https://ridewithvia.com/>

 Google Maps Platform

OverviewProductsPricingDocumentation


Web > Maps JavaScript API

GUIDESREFERENCESAMPLESSUPPORT

Overview☆☆☆☆

The Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries.

<https://developers.google.com/maps/documentation/javascript/tutorial>

 Google Maps Platform

OverviewProductsPricingDocumentation

Web Services > Directions API


GUIDESSUPPORT

Before you begin

This document is intended for website and mobile developers who want to compute direction data within maps provided by one of the Google Maps APIs. It provides an introduction to using the API and reference material on the available parameters.

Before you start developing with the Directions API, review the authentication requirements (you need an API key) and the API usage and billing information (you need to enable billing on your project).

<https://developers.google.com/maps/documentation/directions/intro>

 Google Maps Platform

OverviewProductsPricingDocumentation

Web > Maps JavaScript API

GUIDESREFERENCESAMPLESSUPPORT

Directions Requests

Accessing the Directions service is asynchronous, since the Google Maps API needs to make a call to an external server. For that reason, you need to pass a callback method to execute upon completion of the request. This callback method should process the result(s). Note that the Directions service may return more than one possible itinerary as an array of separate `routes[]`.

<https://developers.google.com/maps/documentation/javascript/directions>

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

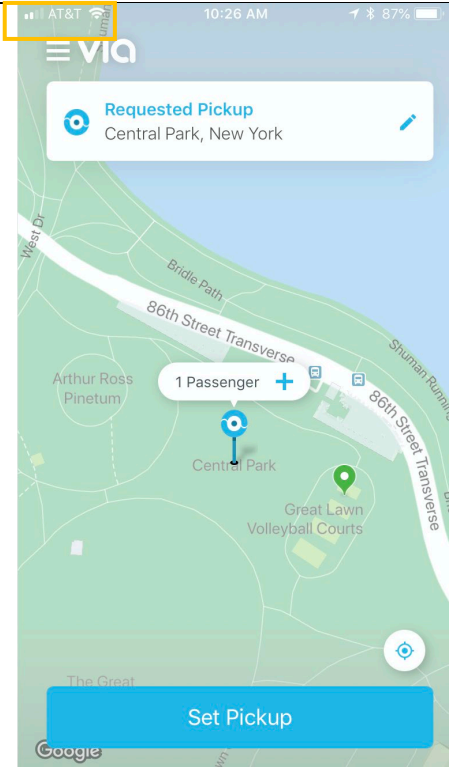
	<div></div> <div>Screenshot from User’s iPhone 6s using Via App</div>
establishing a wireless connection with the client;	<div></div> <div>Screenshot from User’s iPhone 6s using Via App</div>

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

receiving at the server start and end route designations from the client;

AT&T

10:26 AM

87%

Requested Pickup

Central Park, New York

1 Passenger

Set Pickup

Google Maps Platform

Overview

Products

Pricing

Documentation

Web

Maps JavaScript AP

GUIDES

REFERENCE

SAMPLES

SUPPORT

Overview

You can calculate directions (using a variety of methods of transportation) by using the `DirectionsService` object. This object communicates with the Google Maps API Directions Service which receives direction requests and returns an efficient path. Travel time is the primary factor which is optimized, but other factors such as distance, number of turns and many more may be taken into account. You may either handle these directions results yourself or use the `DirectionsRenderer` object to render these results.

When specifying the origin or destination in a directions request, you can specify a query string (for example, "Chicago, IL" or "Darwin, NSW, Australia"), a `LatLng` value, or a `google.maps.Place` object.

The Directions service can return multi-part directions using a series of waypoints. Directions are displayed as a polyline drawing the route on a map, or additionally as a series of textual description within a `<div>` element (for example, "Turn right onto the Williamsburg Bridge ramp").

<https://developers.google.com/maps/documentation/javascript/directions>

Below is a sample `DirectionsRequest` for driving directions:

```
{
  origin: 'Chicago, IL',
  destination: 'Los Angeles, CA',
  travelMode: 'DRIVING',
  drivingOptions: {
    departureTime: new Date(Date.now() + N), // for the time N milliseconds from now.
    trafficModel: 'optimistic'
  }
}
```

<https://developers.google.com/maps/documentation/javascript/directions>

Google Maps Platform

Overview

Products

Pricing

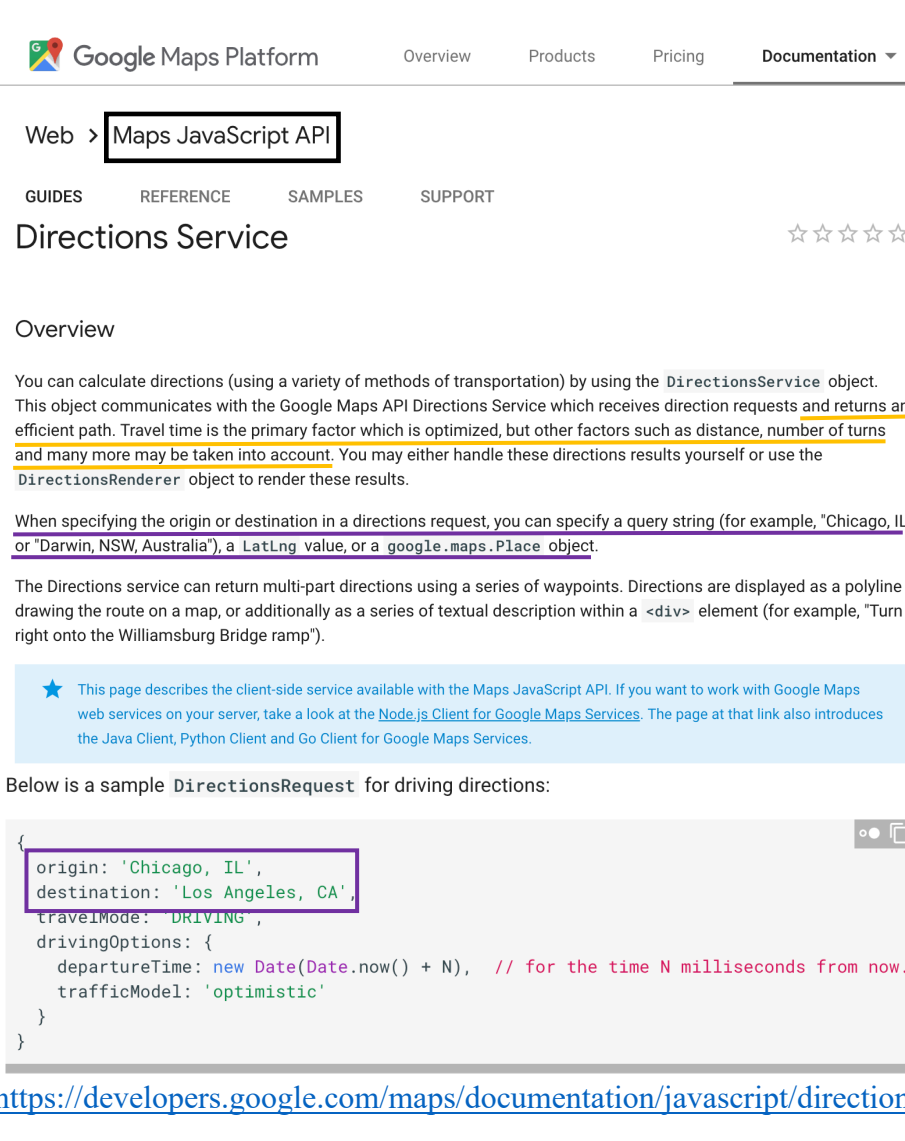
Documentation

Web Services

Directions API

GUIDES

SUPPORT

	<h2>Directions Requests</h2> <p>A Directions API request takes the following form:</p> <pre>https://maps.googleapis.com/maps/api/directions/outputFormat?parameters</pre> <p>where <code>outputFormat</code> may be either of the following values:</p> <ul style="list-style-type: none"><li><code>json</code> (recommended) indicates output in JavaScript Object Notation (JSON)</li><li><code>xml</code> indicates output as XML</li></ul> <h3>Required parameters</h3> <ul style="list-style-type: none"><li><code>origin</code> – The address, textual latitude/longitude value, or place ID from which you wish to calculate directions.</li><li><code>destination</code> – The address, textual latitude/longitude value, or place ID to which you wish to calculate directions. The options for the <code>destination</code> parameter are the same as for the <code>origin</code> parameter, described above.</li></ul> <h3>Example Directions Requests</h3> <p>The following request returns driving directions from Toronto, Ontario to Montreal, Quebec.</p> <pre>https://maps.googleapis.com/maps/api/directions/json?origin=Toronto&amp;destination=Montreal&amp;key=</pre> <p><a href="https://developers.google.com/maps/...DirectionsResponses">https://developers.google.com/maps/...DirectionsResponses</a></p>
<p>calculating at the server the optimal route based on real-time information at the server and said start and end route designations;</p>	 <p>The screenshot shows the Google Maps Platform documentation page for the Directions Service. The page is titled "Directions Service" and includes a navigation bar with links to Overview, Products, Pricing, and Documentation. The main content area is titled "Overview" and describes the DirectionsService object, which communicates with the Google Maps API Directions Service to receive direction requests and return an efficient path. It mentions that travel time is the primary factor optimized, but other factors like distance, number of turns, and many more may be taken into account. It also notes that users can either handle the results themselves or use the DirectionsRenderer object to render them. A note specifies that when specifying the origin or destination, users can use a query string (e.g., "Chicago, IL" or "Darwin, NSW, Australia"), a LatLng value, or a google.maps.Place object. The page also mentions that the Directions service can return multi-part directions using a series of waypoints. A blue callout box states that the page describes the client-side service available with the Maps JavaScript API and provides links to the Node.js Client, Java Client, Python Client, and Go Client for Google Maps Services. Below the callout, a sample DirectionsRequest for driving directions is shown in a code block. The code block contains a JSON object with the following properties: origin: 'Chicago, IL', destination: 'Los Angeles, CA', travelMode: 'DRIVING', drivingOptions: { departureTime: new Date(Date.now() + N), // for the time N milliseconds from now. trafficModel: 'optimistic' }, and a closing brace. The code is highlighted with a purple box. At the bottom of the screenshot, the URL <a href="https://developers.google.com/maps/documentation/javascript/directions">https://developers.google.com/maps/documentation/javascript/directions</a> is visible.</p>

formatting at the  
server the optimal  
route into a non-  
proprietary, natural  
language  
description;



Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

Directions Steps

A `DirectionsStep` is the most atomic unit of a direction's route, containing a single step describing a specific, single instruction on the journey. E.g. "Turn left at W. 4th St." The step not only describes the instruction but also contains distance and duration information relating to how this step relates to the following step. For example, a step denoted as "Merge onto I-80 West" may contain a duration of "37 miles" and "40 minutes," indicating that the next step is 37 miles/40 minutes from this step.

When using the Directions service to search for transit directions, the steps array will include additional `Transit Specific Information` in the form of a `transit` object. If the directions include multiple modes of transportation, detailed directions will be provided for walking or driving steps in a `steps[]` array. For example, a walking step will include directions from the start and end locations: "Walk to Innes Ave & Fitch St". That step will include detailed walking directions for that route in the `steps[]` array, such as: "Head north-west", "Turn left onto Arelious Walker", and "Turn left onto Innes Ave".

The `DirectionsStep` is an object literal with the following fields:

- `instructions` contains instructions for this step within a text string.
- `distance` contains the distance covered by this step until the next step, as a `Distance` object. (See the description in `DirectionsLeg` above.) This field may be undefined if the distance is unknown.
- `duration` contains an estimate of the time required to perform the step, until the next step, as a `Duration` object. (See the description in `DirectionsLeg` above.) This field may be undefined if the duration is unknown.
- `start_location` contains the geocoded `LatLng` of the starting point of this step.
- `end_location` contains the `LatLng` of the ending point of this step.
- `polyline` contains a single `points` object that holds an `encoded polyline` representation of the step. This polyline is an approximate (smoothed) path of the step.
- `steps[]` a `DirectionsStep` object literal that contains detailed directions for walking or driving steps in transit directions. Sub-steps are only available for transit directions.
- `travel_mode` contains the `TravelMode` used in this step. Transit directions may include a combination of walking and transit directions.
- `path` contains an array of `LatLngs` describing the course of this step.
- `transit` contains transit specific information, such as the arrival and departure times, and the name of the transit line.

<https://developers.google.com/maps/documentation/javascript/directions>

Comment:

The optimum route is described in a non-proprietary manor as follows.

See how it works

Google Maps Platform integrates seamlessly with iOS, Android, and desktop applications. [Learn more.](#)

GET DRIVING DIRECTIONS FROM A TO B

ESTIMATE TRAVEL TIME AND DISTANCE

CREATE A ROUTE WITH UP TO 5 WAYPOINTS

Request

`origin:` 75 9th Ave, New York, NY  
`destination:` MetLife Stadium Dr East Rutherford, NJ 07073  
`mode:` driving  
`key:` API\_KEY

URL

`https://maps.googleapis.com/maps/api/directions/json?origin=75+9th+Ave+New+York,+NY&destination=MetLife+Stadium+1+MetLife+Stadium+Dr+East+Rutherford,+NJ+07073&key=YOUR_API_KEY`

Response

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJyYfhZ79ZwokRMtXcL6CYxkA",
      "types" : [ "premise" ]
    },
    {
      "geocoder_status" : "OK",
      "partial_match" : true,
      "place_id" : "ChIJ8YWMWnz4wokRCOVf1CcJCbY",
      "types" : [ "street_address" ]
    }
  ],
  "routes" : [
```

8

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

	<pre>{   "bounds" : {     "northeast" : {       "lat" : 40.8171321,       "lng" : -73.99449150000001     },     "southwest" : {       "lat" : 40.7416627,       "lng" : -74.0728354     }   },   "copyrights" : "Map data ©2015 Google",   "legs" : [     {       "distance" : {         "text" : "9.7 mi",         "value" : 15653       },       "duration" : {         "text" : "25 mins",         "value" : 1480       },       "end_address" : "1 MetLife Stadium Dr, East Rutherford, NJ 07073, USA",       "end_location" : {         "lat" : 40.814505,         "lng" : -74.07272910000002       },       "start_address" : "75 Ninth Ave, New York, NY 10011, USA",       "start_location" : {         "lat" : 40.7428759,         "lng" : -74.00584719999999       },       "steps" : [         {           "distance" : {             "text" : "440 ft",             "value" : 134           },           "duration" : {             "text" : "1 min",             "value" : 34           },           "end_location" : {             "lat" : 40.7422925,             "lng" : -74.004457           },           "html_instructions" : "Head \u003cb\u003esoutheast\u003c/b\u003e on \u003cb\u003eW 16th St\u003c/b\u003e toward \u003cb\u003eNinth Ave\u003c/b\u003e",           "polyline" : {             "points" : "_rtwFpgubMtBuG"           },           "start_location" : {             "lat" : 40.7428759,             "lng" : -74.00584719999999           },         },       ],     },   ], }</pre>
--	--



Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

	<p>Displaying the <b>DirectionsResult</b></p> <p>The <b>DirectionsResult</b> contains the result of the directions query, which you may either handle yourself, or pass to a <b>DirectionsRenderer</b> object, which can automatically handle <b>displaying the result on a map.</b></p> <p>To display a <b>DirectionsResult</b> using a <b>DirectionsRenderer</b>, you simply need to do the following:</p> <ol style="list-style-type: none"><li>1. Create a <b>DirectionsRenderer</b> object.</li><li>2. Call <b>setMap()</b> on the renderer to <b>bind it to the passed map.</b></li><li>3. Call <b>setDirections()</b> on the renderer, passing it the <b>DirectionsResult</b> as noted above. Because the renderer is an <b>MVCObject</b>, it will automatically detect any changes to its properties and <b>update the map</b> when its associated directions have changed.</li></ol> <p>The following example calculates directions between two locations on Route 66, where the origin and destination are set by the given "start" and "end" values in the dropdown lists. The <b>DirectionsRenderer</b> handles <b>display of the polyline between the indicated locations, and the placement of markers at the origin, destination, and any waypoints, if applicable.</b></p> <p><a href="https://developers.google.com/maps/documentation/javascript/directions">https://developers.google.com/maps/documentation/javascript/directions</a></p> <div data-bbox="597 803 1341 1446"></div> <p><a href="https://platform.ridewithvia.com/">https://platform.ridewithvia.com/</a></p>
<p>and display said optimal route on a display system coupled to the client.</p>	<div data-bbox="493 1572 1437 1757"></div> <p>Displaying the <b>DirectionsResult</b></p> <p>The <b>DirectionsResult</b> contains the result of the directions query, which you may either handle yourself, or pass to a <b>DirectionsRenderer</b> object, which <b>can automatically handle displaying the result on a map.</b></p> <p>To display a <b>DirectionsResult</b> using a <b>DirectionsRenderer</b>, you simply need to do the following:</p> <ol style="list-style-type: none"><li>1. Create a <b>DirectionsRenderer</b> object.</li><li>2. Call <b>setMap()</b> on the renderer to bind it to the passed map.</li><li>3. Call <b>setDirections()</b> on the renderer, passing it the <b>DirectionsResult</b> as noted above. Because the renderer is an <b>MVCObject</b>, it will automatically detect any changes to its properties and update the map when its associated directions have changed.</li></ol> <p>The following example calculates directions between two locations on Route 66, where the origin and destination are set by the given "start" and "end" values in the dropdown lists. The <b>DirectionsRenderer</b> handles display of the polyline between the indicated locations, and the placement of markers at the origin, destination, and any waypoints, if applicable.</p> <p><a href="https://developers.google.com/maps/documentation/javascript/directions">https://developers.google.com/maps/documentation/javascript/directions</a></p>

Exhibit A - U.S. PATENT NO. 6,292,743 – MOBILE NAVIGATION SYSTEM

